**Home Display**

Imports BAL

Imports DAL

Imports System.ComponentModel

Public Class HomeDisplay

Private Sub Display\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

HomeScreen(Today)

End Sub

Public Function HomeScreen(ByVal Startupdate As Date)

Try

Dim clinicianinfo As New Clinicians

Dim dsclinicians As New DataSet

REM inactive clinicans are clinician who have left the institute and will not be coming back in the future.

REM Active clinicians are those clinicians who are currently working or are not working for an extended peiod of time.

REM Return all active Clinicians. Passing TRUE value looks for all active clinicians and returns all active clinicians.

REM If one were to choose false it would return all inactive clinicians

dsclinicians = clinicianinfo.GetClinicianInfo(True)

Dim dtClinicians As DataTable = dsclinicians.Tables("Clinician")

Dim row As DataRow

REM Populate columnheaders of the DataGridView control with all the Clinicians that have returned as active

For Each row In dtClinicians.Rows

Dim ClinicianName As String = row("LastName").Trim & ", " & row("FirstName").Trim

DataGridView1.Columns.Add(ClinicianName, ClinicianName)

Next

REM Gererate DataGrid Dispaly Layout, and Populate it

DisplaySetup(Startupdate)

CancelationDisplay()

DataGridView1.SelectionMode = DataGridViewSelectionMode.CellSelect

DataGridView1.AllowDrop = True

DataGridView1.AllowUserToResizeColumns = True

DisplayAttendanceSchedule()

ColorCodeNoShow()

Catch ex As Exception

Throw ex

End Try

Return (Nothing)

End Function

Public Function RemoveColumns()

Me.DataGridView1.Columns.Clear()

Return Nothing

End Function

Private Sub MonthCalendar1\_DateChanged(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DateRangeEventArgs) Handles MonthCalendar1.DateChanged

Dim Schedule\_Date As Date

Schedule\_Date = MonthCalendar1.SelectionStart

Removerows(Schedule\_Date)

CancelationDisplay()

DisplayAttendanceSchedule()

ColorCodeNoShow()

End Sub

REM Setup the column that list all 25 time intervals Between 7:30 AM to 6:00 PM

REM Dataset ds will store all the students in their respective time slots.

REM Dataset ds will store all the students in their respective time slots.

REM Get Selected Calendar value from the CalendarControl

REM The line below is necessary in order to attain other attributes associated with each student such as classroom, and Campus information

REM Return each clincian

REM Iterate through each Clinician listed in each header column of the DataGrid

REM Check to see if the current Clinician matches the Clinician name Listed in the Header Column of the DataGrid control

REM iterate through all the rows/labeled time intervals

REM Store current time calue from the first column of the GridView Control

REM Get the first student name at this time interval that is stored in the dataset

REM Check to see if clinician is off at the particular time interval by testing for an empty string.

REM Otherwise convert the students name to an thier identification number and place him in the respective cell in the Gridview

REM get studentid

REM return additional information of the student

REM Check Proposed(not rescheduled) hours that are hour 1 or 2

REM Check rescheduled hours that are hour 1 or 2

REM Determine the Color which corresponds to the students Status Reschedule/Transfer,Proposed,

REM Also Determine the students Attendance,No Show/Absent then set the respective cell color

REM Also check if there will be a meeting, or Testing

REM reset the studentid and location

REM Check for the first two hours of the current student. If the first two hours are found the flag it inthevariable called start hour

REM Check to see if the first two hours are triggered. If so then color the cell white.

REM check to see if this is a different student from the last iteration.

REM Otherwise the location will show up in every cell/timeslot that corresponds to the student in the current iteration

REM rather than once.

REM Also check to see if student name appears on every hour that is scheduled.

REM store the student name so the current student in the iteration only appears once.

Public Sub DisplaySetup(ByVal CurrentDate As Date)

Try

Dim Student As String = Nothing

Dim Display As New ScheduleConfig

Dim Stat As New Schedule

REM Setup the column that list all 25 time intervals Between 7:30 AM to 6:00 PM

DisplayTemplate()

Dim ds As New DataSet

REM The GridView control will mirror the dataset.

REM Dataset ds will store all the students in their respective time slots.

ds = Display.SetupScheduleColRows(CurrentDate)

Dim dt As DataTable = ds.Tables("Schedule1")

Dim ds2 As New DataSet

Dim ds3 As New DataSet

Dim ds5 As New DataSet

Dim studentid As String

Dim splitname\_student() As String

Dim nextname As String = Nothing

Dim location As String = Nothing

REM Get Selected Calendar value from the CalendarControl

Dim startdate As DateTime = MonthCalendar1.SelectionStart

Dim FinalDate As DateTime = MonthCalendar1.SelectionStart

REM The line below is necessary in order to attain other attributes associated with each student such as classroom, and Campus information

ds2 = Stat.GetSchedule(startdate, FinalDate)

Dim dt2 As DataTable = ds2.Tables("MainSchedule")

Dim b As Integer

Dim Status As String = String.Empty

Dim Subject As String = String.Empty

Dim present As String = String.Empty

Dim Clinician As String = String.Empty

Dim ClinicianHeader As String = String.Empty

Dim countid As String = String.Empty

REM Return each clincian

For Each rw In dt.Rows

Clinician = rw("Clinician")

Dim numberofColumns As Integer = DataGridView1.ColumnCount

Dim headercount As Integer = 0

REM Iterate through each Clinician listed in each header column of the DataGrid

For headercount = 1 To numberofColumns

ClinicianHeader = DataGridView1.Columns(headercount).HeaderText

REM Check to see if the current Clinician matches the Clinician name Listed in the Header Column of the DataGrid control

If ClinicianHeader = Clinician Then

Dim a As Integer = 0

Dim timein As String = String.Empty

Dim t1 As DateTime = Nothing

Dim query As String = String.Empty

Dim c As Integer = 0

REM iterate through all the rows/labeled time intervals

For a = 1 To 24

REM Store current time calue from the first column of the GridView Control

timein = DataGridView1.Rows(a).Cells(0).Value

t1 = Convert.ToDateTime(timein.Trim).ToShortTimeString

REM Get the first student name at this time interval that is stored in the dataset

Student = rw(a).ToString

REM Check to see if clinician is off at the particular time interval by testing for an empty string.

REM Otherwise convert the students name to an thier identification number and place him in the respective cell in the Gridview

If Student = " " & "OUT" Then

location = String.Empty

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Green

If c = 0 Then

DataGridView1.Rows(a).Cells(headercount).Value = " " & "OUT"

DataGridView1.Columns(headercount).DefaultCellStyle.Font = New Font("Times NewRoman", 8, FontStyle.Regular)

End If

c = c + 1

ElseIf Student <> String.Empty Then

splitname\_student = Student.Split(",")

REM get studentid

studentid = Stat.ReturnStudentInfo(splitname\_student(1).Trim, splitname\_student(0).Trim)

Dim timestamp As String = Nothing

timestamp = Convert.ToDateTime("1900-01-01 " & timein)

Dim time1 As DateTime

Dim time2 As DateTime

query = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

REM return additional information of the student

Dim foundrow() As DataRow = dt2.Select(query)

b = 0

Do While b <= foundrow.Length - 1

Status = foundrow(b)("status")

countid = foundrow(b)("Count")

present = foundrow(b)("Attendance")

time1 = foundrow(b)("TimeIn")

time2 = foundrow(b)("TimeOut")

ds5 = Stat.GetClassroomData(countid.Trim)

Dim dt5 As DataTable = ds5.Tables("Classroom")

Dim subjectrow As DataRow

For Each subjectrow In dt5.Rows

Subject = subjectrow("Subject")

location = subjectrow("Campus")

Next

'Check for the first two hours of the current student. If the first two hours are found the flag it inthevariable called start hour

Dim starthr As String = String.Empty

Dim gethr As String = String.Empty

'Check Proposed(not rescheduled) hours that are hour 1 or 2

Dim dt3 As DataTable

Dim query2 = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

ds3 = Stat.ReturnBillingDetailsinfo(studentid, CurrentDate, CurrentDate)

dt3 = ds3.Tables("BillingDetails")

Dim foundrow\_starthour() As DataRow = dt3.Select(query2)

Dim d As Integer = 0

Do While d <= foundrow\_starthour.Length - 1

gethr = foundrow\_starthour(d)("HourNoIn")

If gethr = 1 Or gethr = 2 Then

starthr = gethr

End If

d = d + 1

Loop

'Check rescheduled hours that are hour 1 or 2

Dim transferquery = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

Dim dt4 As DataTable

dt4 = Stat.ReturnStudentTransfer(studentid.Trim, CurrentDate)

Dim foundrow\_transferstarthour() As DataRow = dt4.Select(transferquery)

Dim e As Integer = 0

Do While e <= foundrow\_transferstarthour.Length - 1

gethr = foundrow\_transferstarthour(e)("TransferHrIn")

If gethr = 1 Or gethr = 2 Then

starthr = gethr

End If

e = e + 1

Loop

REM Determine the Color which corresponds to the students Status Reschedule/Transfer,Proposed,

REM Also Determine the students Attendance,No Show/Absent then set the respective cell color

REM Also check if there will be a meeting, or Testing

If present.Trim = "Absent" And (timestamp >= time1 Or timestamp <= time2) Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Red

ElseIf present.Trim = "Completed" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Blue

ElseIf Status.Trim = "Transfer" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Azure

ElseIf Status.Trim = "Proposed" And Subject.Trim = "Testing" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Gray

ElseIf Status.Trim = "Proposed" And Subject.Trim = "Meeting" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Orange

ElseIf present.Trim = "Proposed" And (timestamp >= time1 Or timestamp <= time2) Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Yellow

End If

'Check to see if the first two hours are triggered. If so then color the cell white.

If starthr = "1" Or starthr = "2" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.White

starthr = String.Empty

End If

b = b + 1

Loop

ElseIf Student = String.Empty Then

REM reset the studentid and location

studentid = [String].Empty

location = [String].Empty

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.AntiqueWhite

Else

End If

REM check to see if this is a different student from the last iteration.

REM Otherwise the location will show up in every cell/timeslot that corresponds to the student in the current iteration

REM rather than once.

REM Also check to see if student name appears on every hour that is scheduled.

If nextname <> Student Or nextname = Student And b = 2 Then

If location = "Northwest" Or location = "NorthWest" Then

location = "NW="

ElseIf location.Trim = "Main" Then

location = "M="

Else

location = [String].Empty

End If

REM Concatenate the Location with the student name

DataGridView1.Rows(a).Cells(headercount).Value = location & Student

DataGridView1.Columns(headercount).DefaultCellStyle.Font = New Font("Times NewRoman", 8, FontStyle.Regular)

REM store the student name so the current student in the iteration only appears once.

nextname = Student

End If

Next

Exit For

Else

End If

Next

Next

CancelationDisplay()

DataGridView1.AllowDrop = True

DisplayAttendanceSchedule()

ColorCodeNoShow()

Catch ex As Exception

End Try

End Sub

REM This is a Templat which Populates the all 26 rows in the Firstcolumn of the DataGrid Control with all the time values Between 7:30 AM to 6:00 PM

REM set the Font size and Color

REM Add each row into the DataGrid Control throug interation

Public Sub DisplayTemplate()

REM This is a Templat which Populates the all 26 rows in the Firstcolumn of the DataGrid Control with all the time values Between 7:30 AM to 6:00 PM

REM set the Font size and Color

DataGridView1.Columns(0).DefaultCellStyle.Font = New Font("Times NewRoman", 10, FontStyle.Regular)

Dim i As Integer

Dim timestamp As Array = {"Hour", "7:30 AM", "8:00 AM", "8:30 AM", "9:00 AM", "9:30 AM", "10:00 AM", "10:30 AM", "11:00 AM", "11:30 AM", "12:00 PM", "12:30 PM",

"1:00 PM", "1:30 PM", "2:00 PM", "2:30 PM", "3:00 PM", "3:30 PM", "4:00 PM", "4:30 PM", "5:00 PM", "5:30 PM", "6:00 PM",

"6:30 PM", "7:00 PM", "7:30 PM"}

REM Add each row into the DataGrid Control throug interation

For i = 0 To 25

Dim dgvRow As New DataGridViewRow

Dim dgvCell As DataGridViewCell

dgvCell = New DataGridViewTextBoxCell()

dgvCell.Value = timestamp(i)

dgvRow.Cells.Add(dgvCell)

DataGridView1.Rows.Add(dgvRow)

DataGridView1.Rows(i).Cells(0).Style.BackColor = Color.WhiteSmoke

Next

End Sub

REM Remove every Row in the GridView Control

Public Sub Removerows(ByVal Schedule\_Date As Date)

REM Remove every Row in the GridView Control

Dim w As Integer = DataGridView1.Rows.Count

If DataGridView1.Rows.Count > 0 Then

For b = 0 To DataGridView1.RowCount - 1

DataGridView1.Rows.RemoveAt(0)

Next

End If

DisplaySetup(Schedule\_Date)

End Sub

'Displays rescheduled day for all students on the selected day

'Display only rows which are necessary to be shown

Public Sub CancelationDisplay()

Dim StudentCancelInfo As New BAL.StudentCalendar

Dim Cancelation1 As New DataSet

Cancelation1 = StudentCancelInfo.CancelationData(MonthCalendar1.SelectionStart)

Dim dtCancel As DataTable = Cancelation1.Tables("StudentCalendar")

DataGridView2.DataSource = dtCancel

DataGridView2.Columns(0).Width = 150

DataGridView2.Columns(4).Width = 100

DataGridView2.Columns(1).Width = 100

'Display only rows which are necessary to be shown from the dataset

DataGridView2.Columns(5).Width = 150

DataGridView2.Columns(1).Visible = False

DataGridView2.Columns(2).Visible = False

DataGridView2.Columns(3).Visible = False

DataGridView2.Columns(6).Visible = False

DataGridView2.Columns(7).Visible = False

DataGridView2.Columns(8).Visible = False

DataGridView2.Columns(9).Visible = False

DataGridView2.Columns(10).Visible = False

DataGridView2.Columns(11).Visible = False

DataGridView2.Columns(12).Visible = False

DataGridView2.Columns(13).Visible = False

DataGridView2.Columns(14).Visible = False

DataGridView2.Columns(15).Visible = False

DataGridView2.Columns(16).Visible = False

End Sub

Private Sub ClinicianManagerToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ClinicianManagerToolStripMenuItem.Click

ClinicianConsole.Show()

ClinicianConsole.Focus()

End Sub

Private Sub CloseToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CloseToolStripMenuItem.Click

Me.Close()

End Sub

'Open Clinician Console Form

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton2.Click

ClinicianConsole.Show()

ClinicianConsole.Focus()

End Sub

' Open Student Manager Console Form

Private Sub ToolStripButton1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click

StudentManager.Show()

StudentManager.Focus()

End Sub

Private Sub ModifyScheduleToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Transfer.Show()

Transfer.Focus()

End Sub

'If checkbox is checked or unchecked, Check and uncheck every row.

Private Sub CheckBox1\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged

For i = 0 To DataGridView3.RowCount - 1

If CheckBox1.Checked = True Then

DataGridView3.Rows(i).Cells(0).Value = CheckState.Checked

ElseIf CheckBox1.Checked = False Then

DataGridView3.Rows(i).Cells(0).Value = CheckState.Unchecked

End If

Next

End Sub

'Scan each row for a checkbox that is set to true

'If checkbox is set to true then store each cloumn item into a two dimensional array

'first element of the array represents the row index, and the second number represents the attribute of the column

'increment counter that represents the row index by 1

'Store each label into its respective variable

'Check to see if was previously a Transfer or Proposed

'calculate the time interval between the start time and end time to erase the contents of the text box in the Main Gridview dispaly

'Contents must be set to a in the currently displayed Gridview blank string

'Delete the respective intervals by passing the array of selected row items and the daysof the week

'Display the updated gridview

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

Dim rec As New Updateschedule

Dim convertname As New Schedule

Dim Daysofweek() As String = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}

Dim row As DataGridViewRow = Nothing

Dim totalTime As New ArrayList

Dim appointment As DateTime = Nothing

Dim StartTime As String

Dim EndTime As String

Dim msg As String = Nothing

Dim StudentName As String = Nothing

Dim maxrows As Integer = DataGridView3.Rows.Count

Dim counter As Integer = -1

Dim DeleteDate(maxrows, 4) As String

Dim presence As String = Nothing

Dim splitname\_student() As String

Dim studentid As String = Nothing

Dim currentdate As String = Nothing

'Scan each row for a checkbox that is set to true

For i = 0 To maxrows - 1

'If checkbox is set to true then store each cloumn item into a two dimensional array

'first element of the array represents the row index, and the second number represents the attribute of the column

'increment counter that represents the row index by 1

counter = counter + 1

'Store each label into its respective variable

StudentName = DataGridView3.Rows(i).Cells(1).Value

StartTime = DataGridView3.Rows(i).Cells(2).Value

EndTime = DataGridView3.Rows(i).Cells(3).Value

currentdate = MonthCalendar1.SelectionStart.ToString("M/dd/yyyy")

If StudentName <> "" Then

splitname\_student = StudentName.Split(",")

studentid = convertname.ReturnStudentInfo(splitname\_student(1).Trim, splitname\_student(0).Trim)

Else

End If

If DataGridView3.Rows(i).Cells(0).Value = CheckState.Checked Then

presence = "Completed"

msg = "Present"

ElseIf DataGridView3.Rows(i).Cells(0).Value = CheckState.Unchecked Then

presence = "Absent"

msg = "No Show"

ElseIf DataGridView3.Rows(i).Cells(0).Value = CheckState.Indeterminate Then

'Check to see if was previously a Transfer or Proposed

presence = "Proposed"

End If

DeleteDate(counter, 0) = StudentName

DeleteDate(counter, 1) = MonthCalendar1.SelectionStart.ToString("M/dd/yyyy")

DeleteDate(counter, 2) = StartTime.Trim

DeleteDate(counter, 3) = EndTime.Trim

DeleteDate(counter, 4) = presence.Trim

'calculate the time interval between the start time and end time to erase the contents of the text box in the Main Gridview dispaly

'Contents must be set to a in the currently displayed Gridview blank string

totalTime = rec.Timeintervals\_Calc(StartTime, EndTime)

Next

'Delete the respective intervals by passing the array of selected row items and the daysof the week

'Display the updated gridview

rec.MarkAttendance(DeleteDate, Daysofweek)

Dim Schedule\_Date As Date

Schedule\_Date = MonthCalendar1.SelectionStart

Removerows(Schedule\_Date)

DisplayAttendanceSchedule()

MsgBox("Students have been marked")

End Sub

'An alternative view screen for the user to view student data.

'Populates the display with data of all the students listed on the day selected

Public Sub DisplayAttendanceSchedule()

'Populates the display with data of all the students listed on the day selected

Dim StudentSchedule As New ScheduleConfig

Dim dt1 As DateTime

dt1 = Me.MonthCalendar1.SelectionStart

Dim ds As DataSet

ds = StudentSchedule.ReturnallStudentDailySchedule(dt1, dt1)

Dim dt As DataTable = ds.Tables("StudentCalendar")

DataGridView3.DataSource = dt

DataGridView3.Columns(1).Width = 150

DataGridView3.Columns(2).Width = 70

DataGridView3.Columns(3).Width = 70

DataGridView3.Columns(1).ReadOnly = True

DataGridView3.Columns(2).ReadOnly = True

DataGridView3.Columns(3).ReadOnly = True

DataGridView3.Columns(4).Visible = False

DataGridView3.Columns(5).Visible = False

DataGridView3.Columns(6).Visible = False

DataGridView3.Columns(7).Visible = False

DataGridView3.Columns(8).Visible = False

DataGridView3.Columns(9).Visible = False

DataGridView3.Columns(10).Visible = False

DataGridView3.Columns(11).Visible = False

DataGridView3.Columns(12).Visible = False

DataGridView3.Columns(13).Visible = True

DataGridView3.Columns(14).Visible = False

DataGridView3.Columns(16).Visible = False

DataGridView3.Columns(17).Visible = False

DataGridView3.Sort(DataGridView3.Columns(1), ListSortDirection.Ascending)

MarkIntermediate(dt)

End Sub

'Check for empty DatagridView

'Display the status of each student in the Attendance Grid. Then color each row according their status in the database.

'Proposed=Yellow

'Completed=Blue

'Meeting=Orange

Private Sub MarkIntermediate(ByVal dt As DataTable)

'Check for empty DatagridView

Dim present As String = Nothing

Dim emptygrid As Integer = 0

emptygrid = dt.Rows.Count - 1

If emptygrid < 0 Then

Exit Sub

End If

Dim name, StartTime, EndTime, GridName, GridT1, GridT2, status, subject As String

For i = 0 To DataGridView3.RowCount - 1

DataGridView3.Rows(i).Cells(0).Value = CheckState.Indeterminate

Next

'Dipslay the status of each student in the Attendance Grid. Then color each row according their status in the database.

'Proposed=Yellow

'Completed=Blue

'Meeting=Orange

'

For a = 0 To DataGridView3.RowCount - 1

Dim row As DataRow

For Each row In dt.Rows

name = row("Student Name")

StartTime = row("Start")

EndTime = row("Finish")

status = row("State")

present = row("Attendance")

subject = row("Subject").ToString

GridName = DataGridView3.Rows(a).Cells(1).Value

GridT1 = DataGridView3.Rows(a).Cells(2).Value

GridT2 = DataGridView3.Rows(a).Cells(3).Value

If name = GridName And StartTime = GridT1 And EndTime = GridT2 Then

If status.Trim = "Proposed" And subject.Trim = "Testing" And present = "Proposed" Then

DataGridView3.Rows(a).Cells(0).Value = CheckState.Indeterminate

DataGridView3.Rows(a).DefaultCellStyle.BackColor = Color.Gray

ElseIf present.Trim = "Completed" Then

DataGridView3.Rows(a).Cells(0).Value = CheckState.Checked

DataGridView3.Rows(a).DefaultCellStyle.BackColor = Color.Blue

ElseIf present.Trim = "Absent" Then

DataGridView3.Rows(a).Cells(0).Value = CheckState.Unchecked

DataGridView3.Rows(a).DefaultCellStyle.BackColor = Color.Red

ElseIf status = "Transfer" And present = "Proposed" Then

DataGridView3.Rows(a).Cells(0).Value = CheckState.Indeterminate

DataGridView3.Rows(a).DefaultCellStyle.BackColor = Color.Azure

ElseIf status.Trim = "Proposed" And subject.Trim = "Meeting" Then

DataGridView3.Rows(a).Cells(0).Value = CheckState.Indeterminate

DataGridView3.Rows(a).DefaultCellStyle.BackColor = Color.Orange

ElseIf status.Trim = "Proposed" And present = "Proposed" Then

DataGridView3.Rows(a).Cells(0).Value = CheckState.Indeterminate

DataGridView3.Rows(a).DefaultCellStyle.BackColor = Color.Yellow

End If

End If

Next

Next

End Sub

Public Sub ColorCodeNoShow()

Dim w As Integer = DataGridView3.Rows.Count - 1

For a = 0 To w

If DataGridView3.Rows(a).Cells(0).Value = CheckState.Unchecked Then

DataGridView3.Rows(a).DefaultCellStyle.BackColor = Color.Red

End If

Next

End Sub

Public Sub Completed()

Dim w As Integer = DataGridView3.Rows.Count - 1

For a = 0 To w

If DataGridView3.Rows(a).Cells(0).Value = CheckState.Unchecked Then

DataGridView3.Rows(a).DefaultCellStyle.BackColor = Color.Blue

End If

Next

End Sub

Private Sub StudentManagerToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StudentManagerToolStripMenuItem1.Click

StudentManager.Show()

StudentManager.Focus()

End Sub

Private Sub StudentManagerToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StudentManagerToolStripMenuItem.Click

signin.Show()

signin.Focus()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

RescheduleDailyDisplay.Show()

RescheduleDailyDisplay.Focus()

End Sub

Private Sub Timer1\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

Dim sourceindex = DataGridView1.FirstDisplayedScrollingRowIndex

DataGridView1.FirstDisplayedScrollingRowIndex = sourceindex

End Sub

Private Sub DailyScheduleToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DailyScheduleToolStripMenuItem.Click

PrintofficeSchedules()

End Sub

'Load up Student Calendar form

Private Sub StudentScheduleToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StudentScheduleToolStripMenuItem.Click

StudentCalendar.Show()

StudentCalendar.Focus()

End Sub

'Load up Student Calendar form

Private Sub StudentScheduleToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StudentScheduleToolStripMenuItem1.Click

StudentCalendar.Show()

StudentCalendar.Focus()

End Sub

'Load up print daily office schedule form

Private Sub DailyScheduleToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DailyScheduleToolStripMenuItem1.Click

PrintofficeSchedules()

End Sub

'Display Reschhedule dispaly form. To move a student around within the same day

Private Sub ToolStripMenuItem1\_Click(sender As System.Object, e As System.EventArgs) Handles ToolStripMenuItem1.Click

RescheduleDailyDisplay.Show()

RescheduleDailyDisplay.Focus()

End Sub

'Load up print daily office schedule form

Public Sub PrintofficeSchedules()

PrintScheduleForms.Show()

PrintScheduleForms.Focus()

End Sub

End Class